CLAIMS

What is Claimed is:

1. An apparatus for controlling operations by a client on a stored file, said apparatus comprising:

a first memory associated with the file, said first memory for storing a fixed file security status, said fixed file security status being of a first type;

a second memory associated with the file, said second memory for storing an active file security status, said active file security status initially copied from said fixed file security status and initially being of said first type and changeable to a second type;

a request handler receiving a request from the client to perform operations on the file, said request handler disallowing the client from performing operations on the file if said active file security status is of said first type and allowing the client to perform operations on the file if said active file security status is of said second type; and

a independent verification routine having access to a security database listing clients and corresponding privileges, and capable of receiving a authorization credential from the client, said independent verification routine causing said active file security status to change to said second type if said authorization credential indicates that the client has the privilege to access the file.

2. The apparatus of claim 1, further comprising a third memory associated with the file, said third memory for storing a delete-on-close status, said delete-on-close status initially set to a first value and changeable to a second value,

wherein said first value indicates that the file will not be deleted upon closing and the second type indicates that the file will be deleted upon closing.

3. The apparatus of claim 2, wherein said first memory is a non-volatile random-access memory and said second memory and third memory are in a file entry.

4. The apparatus of claim 3, wherein said first memory, said second memory, and said third memory comprise single bits.

5. A method for controlling operations by a client on a file stored on an apparatus, said apparatus having a first memory associated with said file, said first memory for storing a fixed file security status of a first type, a second memory associated with said file, said second memory capable of storing an active file security status of a first type and changeable to a second type wherein said first type indicates that operations are not allowed on the file and said second type indicates that operations are allowed on the file, and a independent verification routine, said independent verification routine having access to a security database listing clients and their corresponding privileges and receiving a authorization credential from said client, the method comprising:

copying said first type from said fixed file security status stored in said first memory to said active file security status stored in said second memory;

receiving said authorization credential from said client; and

changing said active file security status stored in said second memory to said second type if said independent verification routine determines that the client has the privilege to access the file.

6. The method of claim 5, wherein the apparatus has a third memory associated with the file, said third memory storing a delete-on-close status, said third memory

initially storing a first value and changeable to a second value wherein said first value indicates that the file will not be deleted upon closing and said second value indicates that the file will be deleted upon closing, further comprising:

receiving a delete-on-close request from said client;

changing said first value to said second value; and

deleting the file upon closing.

7. The method of claim 6, wherein said first memory is an NVRAM and said second memory and said third memory are in a file entry.

8. The method of claim 6, wherein said first memory, said second memory, and said third memory comprise single bits.

9. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for controlling operations by a client on a file stored on an apparatus, said apparatus having a first memory associated with said file, said first memory for storing a fixed file security status of a first type, a second memory associated with said file, said second memory capable of storing an active file security status of a first type and changeable to a second type wherein said first type indicates that operations are not allowed on the file and said second type indicates that operations are allowed on the file, and a independent verification routine, said independent verification routine having access to a security database listing clients and their corresponding privileges and receiving a authorization credential from said client, the method comprising:

copying said first type from said fixed file security status stored in said first memory to said active file security status stored in said second memory;

receiving said authorization credential from said client; and

changing said active file security status stored in said second memory to said second type if said independent verification routine determines that the client has the privilege to access the file.

10. A method for creating a secure file on a file system, the method comprising:

receiving from a user an open for write call for a file that does not exist at the time the call is received;

recognizing that the file does not exist at the time the call is received;

creating a file entry for said file;

receiving from said user an authorization credential;

authenticating the privileges of the user;

recognizing the combination of a user sending an open for write call for a file that does not exist at the time the call is received and an authorization credential that is authenticated; and

creating a secure file.

11. The method of claim 10, further comprising:

setting a memory location associated with the file to a value indicating that the file is a secure file.

12. The method of claim 10, further comprising:

closing said file entry.

13. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for creating a secure file on a file system, the method comprising:

receiving from a user an open for write call for a file that does not exist at the time the call is received;

recognizing that the file does not exist at the time the call is received;

creating a file entry for said file;

receiving from said user an authorization credential;

authenticating the privileges of the user;

recognizing the combination of a user sending an open for write call for a file that does not exist at the time the call is received and an authorization credential that is authenticated; and

creating a secure file.